

Маринич І.А.

Криворізький національний університет

Макогонов А.В.

Криворізький національний університет

ВИЯВЛЕННЯ БПЛА НА ОСНОВІ ТЕХНОЛОГІЙ МАШИННОГО ЗОРУ

У статті досліджується проблема виявлення та уникнення перешкод для безпілотних літальних апаратів (БПЛА) коптерного типу, зосереджуючи увагу на використанні сучасних методів комп'ютерного зору та глибокого навчання для детекції БПЛА в реальному часі. Щоб коректно керувати рухом БПЛА в автономному режимі, необхідно мати інформацію про точку зльоту, посадки та траєкторію руху. Ці дані, разом із даними навігаційної системи (GPS, інерційний блок), використовуються для створення програми управління. Під час польоту БПЛА можуть зустрічатися перешкоди, які необхідно оминати. Ці перешкоди можна поділити на дві групи: ті, що знаходяться на низькій висоті (будівлі, дерева), та ті, що зустрічаються на стандартній висоті польоту (інші БПЛА, птахи, літаки). Ця робота зосереджена на виявленні перешкод другої групи, а саме коптерних БПЛА, які зазвичай мають найменші розміри, що ускладнює їх розпізнавання. Оскільки перешкоди на стандартних висотах польоту, такі як інші дрони, є складними для виявлення, особливо через їхні невеликі розміри та схожість з фоном, у роботі розглядаються сучасні нейромережеві архітектури, включаючи YOLOv10, YOLO-NAS, RT-DETR та інші.

Проведено аналіз датасетів, зокрема DUT Anti-UAV Detection and Tracking, які використовуються для навчання моделей. В статті оцінюється продуктивність зазначених моделей на основі точності виявлення, швидкості та обчислювальних витрат, з метою адаптації їх до умов реального часу для вирішення завдань протидії БПЛА. У роботі також розглянуто можливості оптимізації існуючих моделей, зокрема шляхом зменшення кількості параметрів моделей та використання переднавчених мереж, що дозволяє підвищити швидкодію без втрати точності. Запропоновані методи оптимізації можуть бути застосовані для реальних систем виявлення БПЛА в умовах обмежених обчислювальних ресурсів

Результати досліджень демонструють, що поєднання сучасних методів обробки зображень і алгоритмів глибокого навчання є ефективним підходом до автоматизованого управління БПЛА та уникнення перешкод.

Ключові слова: виявлення БПЛА, машинний зір, нейронні мережі, автоматизація, YOLO, RT-DETR.

Постановка проблеми. Для коректного управління рухом безпілотного літального апарата в автономному режимі необхідно мати точку взлету, точку посадки, та траєкторію. На основі цих даних, а також даних, отриманих від системи навігації (GPS, інерційний блок) можна розробити програму, яка буде керувати машиною. Але необхідно враховувати, що під час польоту можуть зустрічатися різні перешкоди, які необхідно обминати. Вважаємо, що їх можна розділити на дві великі групи:

1) перешкоди, які зустрічаються на невеликій висоті (будівлі, дерева, тощо);

2) перешкоди, які будуть зустрічатися на стандартній висоті польоту (інші бпла, птахи, літаки, гвинтокрили).

Ця робота зосереджена на виявленні другої групи. І першим типом перешкоди для виявлення

було вирішено взяти БПЛА коптерного типу, так як зазвичай вони мають найменші розміри, і, відповідно важчи для розпізнавання. Згідно [1] в системах детекції можна використовувати різні типи сенсорів:

- радары
- електро-оптичні/теплові
- акустичні
- радіочастотні
- людське спостереження.

Неавтоматичні системи, до складу яких входять кінцеві користувачі, які відстежують і підтверджують класифікаційну мітку виявленої цілі, як правило, мають найкращі характеристики щодо можливостей класифікації, але, як правило, мають високі експлуатаційні витрати через навчання персоналу та обслуговування системи. Як електрооптичні, так і тепловізійні камери пропонують

високі можливості класифікації з точною локалізацією та визначенням дальності, коли використовується кілька датчиків. Електрооптичні камери, як правило, дешеві, хоча теплові камери дорожчі, але обидві вони чутливі до параметрів навколишнього середовища. З іншого боку, акустичні датчики, як правило, стійкі до умов навколишнього середовища, але їх обмежений ефективний діапазон робить їх менш поширеним варіантом. Нарешті, радіолокаційні датчики є найпоширенішим рішенням для частини виявлення через точну локалізацію та велику дальність у поєднанні з гідними можливостями класифікації, які працюють незалежно від налаштувань середовища [1]. Однак виявлення дронів за допомогою систем, розроблених для літаків, не є простим, оскільки вони часто використовують методи зменшення небажаного відлуння від невеликих, повільних і низько літаючих об'єктів, що є саме тим, що характеризує дрони [2]. Також необхідно врахувати, що додаткове обладнання збільшує вагу, вартість та відповідно енергоспоживання безпілотного літального апарату. Тому локалізація об'єктів за допомогою вбудованої камери і бортового комп'ютера виглядає перспективним напрямом автоматизації управління БПЛА.

Аналіз останніх досліджень і публікацій. За останні роки сильно розширилися можливості застосування технологій машинного навчання для виявлення об'єктів на зображеннях. Прикладами можуть слугувати моделі сімейства You Only Look Once (YOLO), REGION-BASED CONVOLUTIONAL NEURAL NETWORKS (RCNN), FAST R-CNN, FASTER R-CNN, MASK R-CNN, DETR [3] та інші. В статті [3] приводиться великий порівняльний аналіз цих мереж на датасетах PASCAL VOC 2007, PASCAL VOC 2012, and Microsoft COCO по точності визначення об'єктів на картинках. Але контексті розпізнавання в реальному часі не менш важливими характеристиками є швидкість відповідей, а також кількість ресурсів необхідних для роботи нейромережі (оперативна пам'ять, потужність процесора, кількість дискового простору). В цьому контексті можна відзначити, що RCNN та Fast R-CNN мають доволі повільну швидкість визначення – 40 та 2 секунди відповідно [4]. В той же час Faster R-CNN визначає об'єкт за 0.2 секунди [4]. Багато останніх детекторів на основі глибокого навчання хоч і показують переконливу продуктивність для виявлення великих об'єктів у різноманітних ситуаціях в режимі реального часу. Однак виявлення невеликих об'єктів є складним

завданням через складну природу таких об'єктів. Крім того, складність, присутня на кількох зображеннях через існування дисбалансу переднього/заднього плану та спотворення перспективи, спричинене несприятливою погодою та умовами слабого освітлення, ускладнює точне виявлення малих об'єктів [5].

Проблема виявлення об'єкта визначається шляхом визначення положення об'єкта на заданому зображенні (локалізація об'єкта) і класу, до якого належить кожен об'єкт (класифікація об'єкта). Алгоритми виявлення об'єктів на основі глибокого навчання поділяються на дві основні категорії. Перший клас – це двоетапні алгоритми виявлення об'єктів, типовими для яких є Fast R-CNN, Faster R-CNN, Mask R-CNN та інші. Іншим класом є одноетапні алгоритми виявлення об'єктів, типовими для яких є такі алгоритми, як SSD (Liu et al., 2016), YOLOv3 (Redmon & Farhadi, 2018) і YOLOv4 (Bochkovskiy et al., 2020). Одноступінчастий алгоритм виявлення об'єктів має наскрізні переваги в продуктивності, але є низькою з точки зору точності виявлення порівняно з двоступеневим. Двоетапний алгоритм виявлення об'єктів використовує підхід локалізації з подальшим розпізнаванням, що є кращим, ніж одноетапний з точки зору точності виявлення, але він має нижчу продуктивність у реальному часі [6].

Проте ці базові методи не працюють належним чином при безпосередньому застосуванні до БПЛА виявлення та відстеження. Навіть незважаючи на наявність алгоритмів виявлення, які поступово стають зрілими та комерціалізованими, виявлення малих об'єктів в складному фоні все ще є проблемою. БПЛА часто зливається з складним фоном, з великим шумом і перешкодами [7].

Основною метою роботи [7] було використання існуючих найсучасніших методів виявлення та відстеження для ефективної адаптації та вирішення завдання проти БПЛА на рівні даних і методів. Методи, засновані на глибокому навчанні, вимагають тривалого навчання даних для отримання надійної та точної продуктивності. Тому було запропоновано набір даних у видимому світлі для БПЛА, в тому числі підмножини виявлення та відстеження. Також було перенавчено кілька методів виявлення за допомогою цього навчального набору. В якості методів виявлення було використано Faster-RCNN, Cascade-RCNN і ATSS, які є двоетапними методами, та YOLOX та SSD, які є одноетапними методами.

Такий самий набір даних було використано і в роботі [8]. Автори використовують розши-

рення ефективного рецептивного поля (ERF – Effective Receptive Field) – карт ознак, згенерованих з основи YOLOv6. RepLKNet використовується як основа YOLOv6, яка розгортає великі ядра з глибокою згортою. Відтак, щоб подолати тривалий час виведення RepLKNet, впроваджується новаторська мережа LERFNet. LERFNet використовує розширену згортку на додаток до великих ядер, щоб збільшити ERF і подолати проблеми один одного. Лінійний модуль уваги просторового каналу (LAM) використовується, щоб приділити більше уваги найбільш інформативним пікселям і високофункціональним каналам. LERFNet генерує вихідні карти ознак із великим ефективним рецептивним полем та високим упередженням форми, щоб покращити виявлення дронів різних розмірів у складних сценах. Порівнюються бази RepLKNet і LERFNet для Tiny-YOLOv6, Tiny-YOLOv6, YOLOv5s і Tiny-YOLOv7. У порівнянні з вищезгаданими методами, результати запропонованої моделі демонструють більший баланс між точністю та швидкістю. LERFNet збільшує MAP на 2,8%, водночас значно зменшуючи GFLOP та кількість параметрів у порівнянні з оригінальною основою YOLOv6.

В роботі [9] базуючись на аналізі попередніх робіт, пропонують, що для відстеження БПЛА важливо об'єднати інформацію з видимих зображень RGB та ІЧ. Тому було представлено новий набір даних, Anti-UAV, щоб полегшити дослідження відстеження БПЛА. Anti-UAV містить високоякісні відеоряди високої чіткості як RGB, так і ІЧ. Кожна послідовність анована обмежувальними рамками, атрибутами та прапорцями, які вказують на існування цільового об'єкта. Варто зазначити, що в Anti-UAV відеоряди RGB та ІЧ об'єднані в пару. Таким чином, запропонований набір даних підтримує як одномодальне, так і мультимодальне відстеження БПЛА.

Постановка завдання. У роботі потрібно дослідити сучасні неймережі виявлення БПЛА

використовуючи наявні датасети DUT Anti-UAV Detection and Tracking [11] та розглянути можливості оптимізації існуючих моделей, зокрема шляхом зменшення кількості параметрів моделей та використання переднавчених мереж, що дозволяє підвищити швидкодію без втрати точності.

Виклад основного матеріалу. Датасет DUT Anti-UAV Detection and Tracking містить загалом 10 000 зображень, у яких набори для навчання, тестування та перевірки містять 5200, 2200 та 2600 зображень відповідно. З огляду на ситуацію, що одне зображення містить кілька об'єктів, загальна кількість виявлення об'єктів становить 10109, де проводиться навчання, тестування та перевірка набори мають 5243, 2245 і 2621 об'єктів відповідно [7]. Набір даних містить зображення з різними розширеннями. Найбільше зображення має розмір 3744px і 5616px, тоді як розмір найменшого зображення складає 160×240.

Балансуючи між різноманітністю предметів і запобіганню надмірного ряду моделей, в ньому представлено більше 35 типи БПЛА. Інформація про задній фон в наборі даних також різноманітна. Враховуючи, що БПЛА здебільшого літають на відкритому повітрі, фоном є зовнішнє середовище, включаючи небо, темні хмари, багатопверхівки, житлові будинки, сільськогосподарські угіддя, та дитячі майданчики. Крім того, враховано різні умови освітлення (наприклад день, ніч, світанок і сутінки), а також різна погода (наприклад, сонячний, хмарний і сніжний день).

Найчастіше розміри БПЛА малі і займають невелику частку на зображенні. Тому у цьому наборі об'єкти, які показані крупним планом займають не значну кількість (приклади наведені на рисунку 1).

Середнє співвідношення площі об'єкта до загального розміру зображення становить приблизно 0,013, найменший коефіцієнт площі об'єкта становить 1,9e-06, а найбільший об'єкт займає 0,7

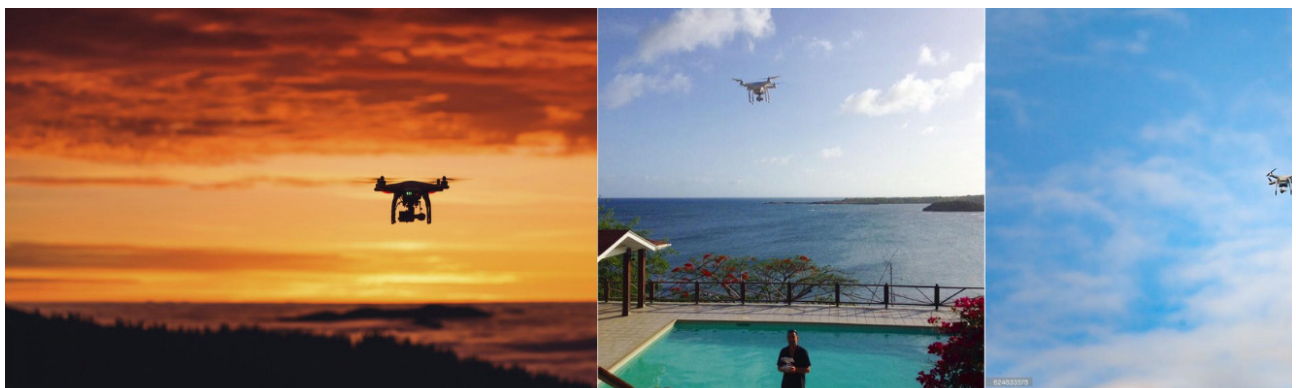


Рис. 1. Приклади зображень з набору даних

від усього зображення. Більшість об'єктів зосереджено в центрі зображення. Діапазон руху об'єктів у всіх наборах різний, а горизонтальні та вертикальні переміщення об'єктів розподілені рівномірно. Для набору даних відстеження обмежувальні рамки об'єкта в одній послідовності є безперервними. Окрім центральної області зображення, об'єкти також часто рухаються праворуч і нижньо-ліворуч від зображення [7].

Для навчання були обрані різні сучасні моделі. У [12] розробники націлившись як на постобробку, так і на архітектуру моделі під час виявлення розробили сімейство моделей YOLOv10. З цією метою спочатку було вирішено проблему надлишкових прогнозів у постобробці шляхом представлення послідовної стратегії подвійних призначень для YOLO без NMS з подвійною міткою призначення та узгоджену метрику відповідності. Це дозволяє усунути потребу в NMS під час виводу (аналізу нових даних), що призводить до конкурентоспроможності з високою ефективністю. По-друге, запропоновано цілісну ефективну керувану стратегію проектування моделі для архітектури моделі шляхом виконання всебічної перевірки для різних компонентів YOLO. Для підвищення ефективності розроблено легку класифікаційну головку, просторово-каналне роз'єднане зменшення розмірності та конструкцію блоків з урахуванням рангу, щоб зменшити виявлену обчислювальну надлишковість та досягти більш ефективної архітектури. Для підвищення точності досліджено згортку з великим ядром і представлено ефективний модуль часткової самоуваги для посилення можливостей моделі, використовуючи потенціал для покращення продуктивності за низьких витрат [12].

Наступною моделлю є YOLO-NAS, яка була випущена у травні 2023 року компанією Deci, яка розробляє моделі та інструменти для створення, оптимізації та розгортання моделей глибокого навчання. YOLO-NAS призначена для виявлення дрібних об'єктів, покращення точності локалізації і підвищення співвідношення продуктивності на обчислювальну машину, що робить її придатною для додатків периферійних пристроїв у режимі реального часу. Крім того, її архітектура з відкритим кодом доступна для дослідницького використання.

Особливостями YOLO-NAS [13] є:

- Модулі з урахуванням квантизації [125], названі QSP та QCI, які поєднують репараметризацію для 8-бітної квантизації, щоб мінімізувати втрату точності під час пост-тренувальної квантизації.

- Автоматичне проектування архітектури за допомогою AutoNAS, власної технології NAS компанії Deci.

- Метод гібридної квантизації для вибіркової квантизації певних частин моделі з метою балансування затримки та точності замість стандартної квантизації, де впливу зазнають усі шари.

- Режим попереднього навчання з автоматично розміченими даними, самодистиляцією та великими наборами даних.

Також останнім часом почала набирати популярність нова архітектура розпізнавання заснована на трансформерах – DETR (Detection Transformer). Вивчаючи дані статті [3] можна дійти висновку, що вони не мають ще достатньої точності, та не можуть бути застосовані для обробки даних у форматі реального часу. Тому автори [14] створили Real-Time DETECTION Transformer (RT-DETR) – перший наскрізний детектор об'єктів у реальному часі. В той час як, YOLO зазвичай вимагають немаксимального придушення (NMS) для постобробки, яка не тільки сповільнює швидкість висновку, але також вводить гіперпараметри, які спричиняють нестабільність як швидкості, так і точності. Крім того, враховуючи те, що різні сценарії роблять різний акцент на запам'ятовуванні та точності, необхідно ретельно вибирати відповідні порогові значення NMS, що перешкоджає розробці детекторів реального часу. DETR напротивагу забезпечує альтернативний підхід:

- Біективне співставлення: DETR створює фіксовану кількість прогнозів і використовує біективне співставлення між прогнозами та істинними об'єктами.

- Навчання end-to-end: Модель навчається безпосередньо видавати унікальні прогнози без потреби в пост-обробці.

- Self-attention механізми: Використовуються для неявного моделювання відносин між об'єктами.

Як результат набір унікальних детекцій видається без застосування NMS. Для зниження обчислювальних витрат цей підхід у [7] був покращений за допомогою ефективного гібридного кодера, щоб замінити кодер Vanilla Transformer, який значно покращує швидкість логічного висновку шляхом роз'єднання внутрішньомасштабної взаємодії та крос-масштабного злиття функцій із різними масштабами. Щоб уникнути вибору функцій кодера з низькою достовірністю локалізації як об'єктних запитів, запропоновано вибір запиту з мінімальною невизначеністю, який забезпечує

високоякісні початкові запити до декодера шляхом явної оптимізації невизначеності, тим самим підвищуючи точність. Крім того, завдяки багаторівневій архітектурі декодера DETR RT-DETR підтримує гнучке налаштування швидкості для адаптації до різних сценаріїв у реальному часі без повторного навчання.

Базуючись на попередніх роботах з цим набором даних та на огляді сучасних рішень для тестування були відібрані декілька архітектур. Їх назви та параметри наведені у таблиці 1. Всі параметри отримані при вхідних зображеннях 640x640px, batch_size =16 за допомогою стандартних функцій фреймворка Ultralytics, за виключенням YOLO-NAS-l (використано бібліотеку thor, тому що стандарта функція фреймворку віддавала значення 0).

Для тренування моделей використовувались інстанси, наведені у таблиці 2.

Приклад однієї з навчальних партій (training batch), яка була використана моделлю YOLOv10s під час тренування наведена на рисунку 2.

Всі моделі проходили цикл тренування у 100 епох. Метриками при тренування моделей YOLO є:

Таблиця 1

Оціночна складність моделей

Назва моделі	Кількість шарів	Кількість параметрів	GFLOPs
YOLOv10n	385	2,775,520	8.7
YOLOv10s	402	8,128,272	25.1
YOLOv10m	498	16,576,768	64.5
YOLOv10l	628	25,888,688	127.9
YOLOv10x	688	31,808,960	171.8
YOLO-NAS-l	719	19,053,888	17.52
RT-DETR Large	673	32,970,476	108.3

Таблиця 2

Параметри серверів для тренування

GPU	Об'єм оперативної пам'яті GPU, Гб	Потужність GPU, TFLOPS	Процесор	Треновані моделі
RTX A4000	16	20.6	Pentium Gold G6400	YOLOv10n, YOLOv10s, YOLOv10m, YOLO-NAS-l
RTX 3090	24	35.3	Core i9-10900X	YOLOv10x, RT-DETR Large



Рис. 2. Приклад training batch для моделі YOLOv10s

- train/box_loss: Втрати при передбаченні обмежувальних рамок (bounding boxes) на тренувальному наборі даних.
- train/cls_loss: Втрати при класифікації об'єктів на тренувальному наборі даних.
- train/df_l_loss: Втрати функції розподілу (Distribution Focal Loss) на тренувальному наборі. Це специфічна втрата, що використовується в деяких версіях YOLO для покращення точності локалізації об'єктів.
- metrics/precision(B): Точність моделі на валідаційному наборі даних. Показує, яка частка передбачених позитивних результатів є дійсно позитивними.
- metrics/recall(B): Повнота моделі на валідаційному наборі даних. Показує, яка частка фактичних позитивних результатів була правильно ідентифікована.
- metrics/mAP50(B): Середня точність (mean Average Precision) при порозі IoU (Intersection over Union) 0.5 на валідаційному наборі.
- metrics/mAP50-95(B): Середня точність, усереднена по різних порогах IoU від 0.5 до 0.95 з кроком 0.05.
- val/box_loss: Втрати при передбаченні обмежувальних рамок на валідаційному наборі даних.
- val/cls_loss: Втрати при класифікації об'єктів на валідаційному наборі даних.
- val/df_l_loss: Втрати функції розподілу на валідаційному наборі.

- lr/pg0, lr/pg1, lr/pg2: Швидкості навчання (learning rates) для різних груп параметрів моделі.

Графічно показники цих метрик це представлені на рисунку 3.

Як видно з графіків основна метрика mAP50 згладжується при наближенні до 100 епохи навчання. При цьому train/box_loss демонструє потенціал до подальшого зменшення, але враховуючи також вирівнювання val/box_loss можна зробити висновок, що збільшення кількості епох може призвести до перенавчання моделі на тренувальному наборі даних.

Модель з трансформерами значно раніше і більш жорстко випрамляє графік (рисунок 4). Тобто можна зробити висновок, що їй треба менше епох для навчання на тому ж самому наборі даних.

Результати розпізнавання об'єктів на валідаційному наборі даних приведено на рисунку 5.

Тестування здійснювалося на сервері Linux з параметрами: 4 ядра процесора AMD EPYC 7282 (загалом ця модель має 16 ядер) та 6 Гб оперативної пам'яті. Тобто на відміну від попередніх робіт, де для перевірки швидкості роботи використовувалися GPU, було обрано конфігурацію без графічного процесора. Яка наближена до конфігурації міні комп'ютера (наприклад Raspberry PI 5).

У таблиці 3 наведені найголовніші метрики для нейромереж, які визначають об'єкти на зображеннях – mAP50 та mAP50-95, а також швидкість обробки.

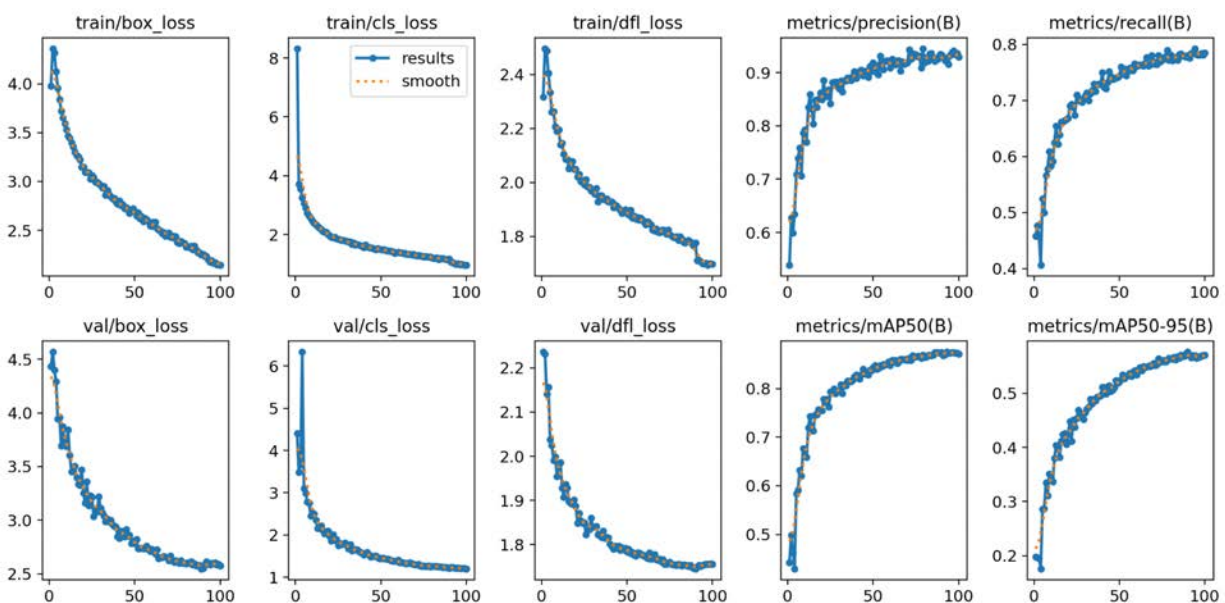


Рис. 3. Графічне відображення метрик навчання для моделі YOLOv10s

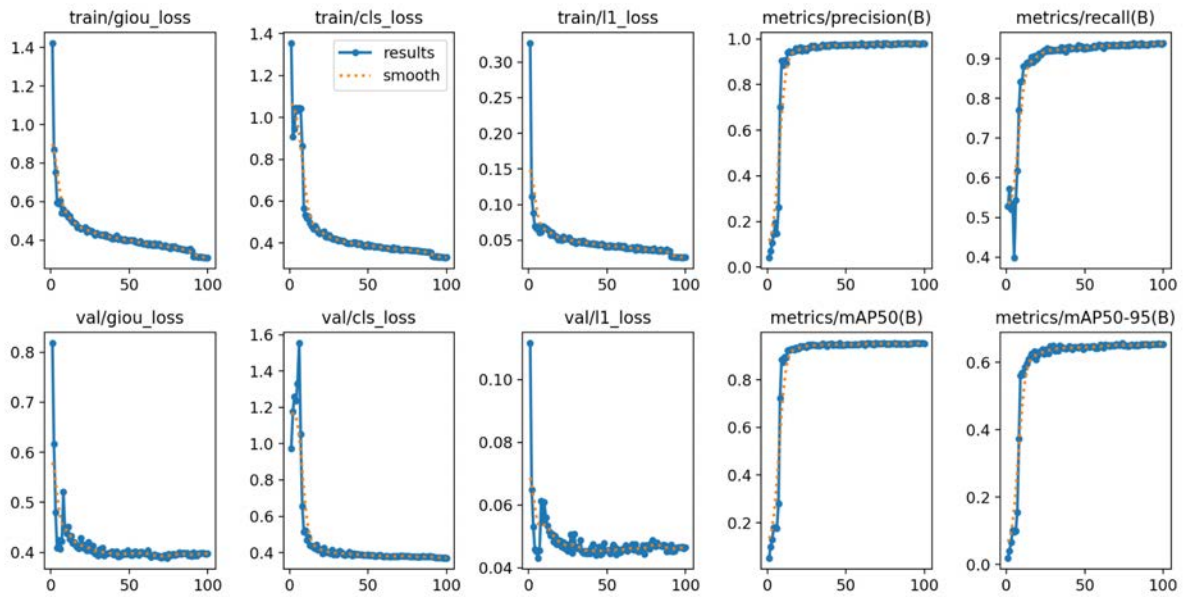


Рис. 4. Графічне відображення метрик навчання для моделі RT-DETR Large



Рис. 5. Приклад розпізнавання БПЛА на валідаційному наборі даних для моделі YOLOv10s

Так як робота націлена на вивчення можливостей використання моделей машинного навчання безпосередньо на борту БПЛА, де скоріш за все не буде встановлена відеокарта.

Останні 3 результати були відібрані з робіт [7] та [8], як найкращі результати. Слід зауважити, що тестування YOLOv6 RepLKNet та YOLOv6 LERFNet відбувалося на інстансі GPU (NVIDIA Quadro M 2000 M) з 4 Гб вбудованої пам'яті. На рахунок Cascade-RCNN не зазначено на якій конфігурації відбувалося тестування. Але в даному

випадку це не є принципово, так як у порівнянні з іншими моделями вона має малу mAP50. Найвищу точність має модель RT-DETR, але, як і очікувалося вона має і найбільшу вартість обчислення. Також можна зробити висновок, що збільшення моделей YOLO не несе сильного приросту у mAP50, але в той же час значно може збільшуватися час на обробку. Вважаємо, що конкурентним вибором у межах малого обчислювального ресурсу доречним буде використання моделі YOLOv10n. При збільшенні потужності, або

Результати навчання моделей на користувацькому наборі даних та їх порівняння до попередніх робіт з цим же датасетом

Модель	mAP50	mAP50-95	Швидкість відповідей, ms	Кадрів в секунду
YOLOv10n	0.902	0.6	58.2	17.18
YOLOv10s	0.919	0.642	122.7	8.15
YOLOv10m	0.92	0.643	304.5	3.28
YOLOv10l	0.92	0.647	528.8	1.89
YOLOv10x	0.929	0.662	807	1.24
YOLO-NAS-l	0.571	0.404	422.5	2.37
RT-DETR Large	0.973	0.717	889.6	1.12
Cascade-RCNN	0.683	-	18.62	53.7
YOLOv6 RepLKNet	93.8%	58.6%	39.20	25.51
YOLOv6 LERFNet	93.5%	58.3%	17.71	56.46

винесенні його на окремий сервер, також варто розглядати варіант з застосуванням RT-DETR.

Висновки. В статті було розглянуті сучасні неймерджі для виявлення об'єктів типу БПЛА квадрокоптерного типу на основі датасету DUT Anti-UAV Detection and Tracking. Також проведено їх навчання та подальше тестування на цьому датасеті і середовищі з обмеженими обчислювальними ресурсами. Додатко було відібрано декілька попередніх робіт з цим датасетом, для того, щоб можна

було впевнитися у точності застосування новий моделей для визначення об'єктів. Найбільш точною виявилася архітектура RT-DETR, запропонована у [14], а найбільш швидкою YOLOv10n (так як вона найлегша з точки зору кількості параметрів). Тому на поточний стан можна обирати ці дві моделі для подальших досліджень, але спираючись на доступні потужності з обчислення. Також в подальшому для виявлення перешкод у повітряному просторі варто збільшувати кількість класів об'єктів.

Список літератури:

1. S. Samaras, E. Diamantidou, D. Ataloglou, N. Sakellariou, A. Vafeiadis, V. Magoulitanis, A. Lalas, A. Dimou, D. Zarpalas, K. Votis, P. Daras, D. Tzovaras. Deep Learning on Multi Sensor Data for Counter UAV Applications – A Systematic Review. *Sensors*. 2019. *Sensors*. Vol. 19. № 22. P. 4837-4872. DOI:10.3390/s19224837.
2. F. Svanström, C. Englund, F. Alonso-Fernandez. Real-Time Drone Detection and Tracking With Visible, Thermal and Acoustic Sensors. *Proc. Intl Conf on Pattern Recognition, ICPR*. (Milan, Italy, 10-15 January 2021). P. 7265-7272. DOI: 10.48550/arXiv.2007.07396
3. A. B. Amjoud, M. Amrouch. Object Detection Using Deep Learning, CNNs and Vision Transformers: A Review. *IEEE Access*. 2023. Vol. 11. P. 35479-35516. DOI: 10.1109/ACCESS.2023.3266093.
4. W. Kadhim, Dr. Mohammed, A. Taha. Object Detection Techniques: A Review. *Wasit Journal of Computer and Mathematics Science*. 2023. № 3. P. 59-68. DOI:10.31185/wjcms.165
5. B. Mahaur, K.K. Mishra. Small-object detection based on YOLOv5 in autonomous driving systems. *Pattern Recognition Letters*. 2023. Vol. 168. P. 115-122. DOI: 10.1016/j.patrec.2023.03.009
6. T. Gao, M. Wushouer, G. Tuerhong. Small Object Detection Method based on Improved YOLOv5" *International Conference on Virtual Reality, Human-Computer Interaction and Artificial Intelligence (VRHCIAI)*, Changsha, China. 2022. P. 144-149. DOI: 10.1109/VRHCIAI57205.2022.00032.
7. J. Zhao, J. Zhang, D. Li, D. Wang. Vision-Based Anti-UAV Detection and Tracking. *Transactions on Intelligent Transportation Systems*. 2022. Vol. 23, № 12. P. 25323-25334. DOI: 10.1109/TITS.2022.3177627.
8. M. Elsayed, M. Reda, A. S. Mashaly, A. S. Ameen. LERFNet: an enlarged effective receptive field backbone network for enhancing visual drone detection. *The Visual Computer*. 2024. DOI: 10.1007/s00371-024-03527-8
9. N. Jiang. Anti-UAV: A Large-Scale Benchmark for Vision-Based UAV Tracking. *Transactions on Multimedia*. 2023. Vol. 25. P. 486-500. DOI: 10.1109/TMM.2021.3128047.
11. J. Zhao. DUT Anti-UAV Detection and Tracking. URL: <https://github.com/wangdongdut/DUT-Anti-UAV> (дата звернення 15.09.2024).
12. A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, G Ding. Yolov10: Real-time end-to-end object detection. *arXiv* 2024. DOI:arXiv:2405.14458
13. J. Terven, D. Cordova-Esparza. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* 2023. Vol. 5. P. 1680-1716. DOI: 10.3390/make5040083

14. Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu, J. Chen. DETRs Beat YOLOs on Real-time Object Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (Seattle WA, USA, 17-21 червня, 2024). 2024. P. 16965-16974. DOI:10.48550/arXiv.2405.14458

Marynych I.A., Makohonov A.V. DETECTION OF UAVS USING MACHINE VISION TECHNOLOGIES

The article explores the problem of detecting and avoiding obstacles for unmanned aerial vehicles (UAVs) of the copter type, focusing on the use of modern computer vision and deep learning methods for real-time UAV detection. To effectively manage the movement of a UAV in autonomous mode, information about the take-off point, landing point, and flight trajectory is essential. This data, along with navigation system data (GPS, inertial sensors), is used to create the control program. During flight, UAVs may encounter obstacles that need to be avoided. These obstacles can be divided into two groups: those at low altitudes (buildings, trees) and those encountered at standard flight altitudes (other UAVs, birds, airplanes). This work focuses on detecting obstacles of the second group, specifically copter UAVs, which are usually smaller in size, making them more difficult to recognize.

Since obstacles at standard flight altitudes, such as other drones, are challenging to detect, particularly due to their small size and similarity to the background, the paper examines modern neural network architectures, including YOLOv10, YOLO-NAS, RT-DETR, and others. The analysis includes datasets, such as the DUT Anti-UAV Detection and Tracking, used to train models. The paper evaluates the performance of these models based on detection accuracy, speed, and computational cost, aiming to adapt them to real-time conditions in UAV countermeasures. The work also explores optimization possibilities for existing models, such as reducing the number of model parameters and using pre-trained networks, which enhances performance without compromising accuracy. The proposed optimization methods can be applied to real-world UAV detection systems with limited computational resources.

The research results demonstrate that the combination of modern image processing methods and deep learning algorithms is an effective approach to automated UAV management and obstacle avoidance.

Key words: UAV detection, machine vision, neural networks, automation, YOLO, RT-DETR.